REMARKS

This Amendment is in response to the Office Action dated March 7, 2006 ("OA"). In the Office Action, claims 1-22 were rejected under 35 USC §§ 101 and 103. Additionally, claims 1, 3, 5, 10, 14 and 17 were rejected under 35 USC §112. By this amendment, claims 1, 3, 5, 10 and 14 are amended. Currently pending claims 1-23 are believed allowable, with claims 1, 3, 5, 10, 14 and 17 being independent claims.

CLAIM REJECTIONS UNDER 35 USC §101:

Claims 1-22 were rejected under 35 USC §101 as directed to non-statutory subject matter. According to the Examiner, claims 1-22 "recite a set of mental processes or the manipulation of an abstract idea." OA, pg. 2.

Claims 1, 3, 5, 10 and 14 are amended herein to recite that the subject matter of each claim is embodied in computer readable medium. The applicants believe that these claim amendments overcome the 35 USC §101 non-statutory subject matter rejections. Claims 2, 4, 6-9, 11-13, 15, 15 and 18-22 are dependent on (directly or indirectly) and further limit one of the amended claims. Thus, claims 2, 4, 6-9, 11-13, 15, 15 and 18-22 are also believed to overcome the 35 USC §101 non-statutory subject matter rejections.

The Applicants respectfully disagree with the Examiner's conclusion that claim 17 recites a set of mental processes or the manipulation of an abstract idea. Claim 17 recites an annotation server is a computer or program that responds to commands from a client. See Computer Dictionary Fifth Edition, Microsoft Press (2002). Thus, claim 17 is not directed to a set of mental processes or the manipulation of an abstract idea.

CLAIM REJECTIONS UNDER 35 USC §112:

Claims 1-22 were rejected under 35 USC §112, first paragraph, as allegedly failing to comply with the enablement requirement. OA, pg. 3. Specifically, the Examiner alleges that the claims contain subject matter that is not described in the specification in such a way as to enable one skilled in the art to make and/or use the invention. OA, pg. 3.

There is a strong presumption that an adequate written description of the claimed invention is present when the application is filed. MPEP 2163 citing In re Wertheim, 541 F.2d 257, 263, 191 USPQ 90, 97 (CCPA 1976) ("we are of the opinion that the PTO has the initial burden of presenting evidence or reasons why persons skilled in the art would not recognize in the disclosure a description of the invention defined by the claims"). "The examiner has the initial burden, after a thorough reading and evaluation of the content of the application, of presenting evidence or reasons why a person skilled in the art would not recognize that the written description of the invention provides support for the claims. There is a strong presumption that an adequate written description of the claimed invention is present in the specification as filed." MPEP 2163 citing Wertheim, 541 F.2d at 262, 191 USPQ at 96.

Claim 1:

Claim 1 recites, in part, "A data processing method embodied in computer readable medium for addressing at least one predetermined element in a structured document, the method comprising the steps of: when the structured document having said at least one predetermined element addressed by predetermined addressing information is modified, inputting the structured document . . ." App., pg. 30, ln. 3-9. The examiner alleges several issues with the cited claim wording.

The Examiner alleges difficulty in "distinguish[ing] between the use of the word 'addressing' as a verb and the word 'address' as a noun." OA, pg. 4. The Applicants respectfully submit that the Examiner's reasoning is purely conclusory without supporting evidence. The Office Action has not explained, and it not evident, why a person of ordinary skill in the art would have found it difficult to distinguish between the uses of the words.

Furthermore, the term "addressing information" is commonly used in the art to refer to meta-information existing for the purpose of facilitating the location of an element of data. "Addressing" will therefore be understood by one skilled in the art as the first half of a two-word noun phrase. By contrast, the word "addressing" is used within the phrase "for addressing", which is clearly a verb. Additionally, the application provides context allowing the audience to understand the meaning of the term "addressing information". It specifies that "XPath is a language for addressing particular parts of a structured document. Using XPath as addressing information allows..." App., pg. 1, ln. 22-24. It follows directly from this citation that addressing information, as used within the application, means information for addressing elements of a structured document.

The Examiner further alleges that the cited claim wording "fails to provide one of ordinary skill in the art with a proficient means of recreating applicant's invention." OA, pg. 4. The Applicants respectfully submit that the Examiner's reasoning is purely conclusory without supporting evidence. The Office Action has not explained, and it not evident, why a person of ordinary skill in the art would have difficulty in recreating the present invention.

Regarding the specific claim wording cited, significant detail is provided concerning how to implement the steps discussed. The cited wording refers to "...inputting the structured document..." App., pg. 30, ln. 8-9. Figure 1 provides numerous details of an exemplary configuration of a computer suitable for implementing the present invention. Specific input devices, including a "network interface" (App., pg. 10, ln. 25), a "USB port" (App., pg. 10, ln. 25-26), a "floppy drive" (App., pg. 10, ln. 27-28), a "CD-ROM (Compact Disc Read Only Memory) driver" (App., pg. 11, ln. 10-11) and a "DVD-ROM (Digital Versatile Disc Read Only Memory) driver" (App., pg. 11, ln. 11-12) are listed. App., pg. 10-11. It would be apparent to one of ordinary skill in the art to use these input devices to input a structured document. Furthermore, one of ordinary skill in the art would be able to input a structured document using any of these input devices. Indeed, the application discusses an embodiment wherein "the program is input via the network interface 106 or the floppy disk drive 109 shown in Figure 1, or a CD-ROM drive (not shown) and stored in the hard disk 105." App., pg. 12, ln. 3-6.

Claim 1 further recites, in part, "...reading the analysis result from the memory; and updating the addressing information according to the analyzed modification so that the addressing information addresses at least one corresponding element..." App., pg. 30, ln. 11-15. The examiner alleges that this wording is "vexatious." OA, pg. 4.

The Applicants respectfully submit that the Examiner's reasoning is purely conclusory without supporting evidence in the cited references. The Office Action has not explained, and it not evident, why a person of ordinary skill in the art would have difficulty understanding the cited claim wording.

Regarding the specific claim wording cited, significant detail is provided concerning how to implement the steps discussed. The application specifies implementation details on how to retrieve "the analysis result from memory". App., pg. 30, ln. 11. The Application states, "The XPath update unit 30 receives inputs to be processed, that is, the unmodified document P, the modified document P', the differences D between the unmodified document P and the modified document P', and the XPath (P) from memory means such as the main memory 103 or the hard disk 105." App., pg. 20, ln. 18-23. Furthermore, numerous implementation details concerning "updating the address information according to the analyzed modification..." (App., pg. 30, ln. 12-13) are provided by Figures 7 through 14 and their related discussion. App., pg. 20-25.

The Examiner further states that the wording "leaves one of ordinary skill in the art oblivious as to what is doing the reading and addressing." The Applicants respectfully submit that no legal requirement exists to specify the actor in a method claim such as Claim 1. Thus, it is irrelevant whether the identity and nature of the actor are specified by the claim.

For at least these reasons, claim 1 is believed allowable. The Applicant respectfully request reconsideration and allowance of claim 1.

Claims 3 and 14:

Claim 3 recites, in part, "...a second step of reading the operation sequences from the memory and changing operations in the operation sequence that are interpreted as a movement of a component into an operation of moving the component." App., pg. 31, ln. 8-11. Claim 14 recites similar wording. App., pg. 35, ln. 1-5. The examiner alleges several issues with the cited claim wording.

The Examiner alleges that the claim does not completely describe the operation. The Examiner further states that "[t]he operation of moving the component leaves one to wonder how the component is being moved and where the component is being moved to." OA, pg. 4. The Applicants respectfully submit that the Examiner's reasoning is purely conclusory without supporting evidence. The Office Action has not explained, and it not evident, why a person of ordinary skill in the art would have found it difficult to understand the nature of the operations.

To the contrary, an understanding of the wording used in Claims 3 and 14 is provided by the application. Specifically, the application clarifies that the term "operation sequences" refers to logical operations to manipulate the structured documents, as opposed to operations performed by an embodiment of the present invention. One example clarifying the word usage is as follows: "...it [the difference computation unit] analyzes operations required for modifying the tree T into the tree T' in terms of the basic operations, RemoveNode, InsertNode, and Modify, and generates a list L of obtained operation sequences." App., pg. 16, ln. 17-20.

The application also provides sufficient detail for one of ordinary skill in the art to recreate the steps specified by the cited claim wording. Specifically, Figure 3 provides a detailed algorithm for combining an operation in which a node (e.g., a component) is inserted with another operation in which the same node is removed into an operation of moving the node. App., pg. 16-17. Likewise, Figure 4 provides a detailed algorithm for combining an operation in which a node is removed with another operation in which the same node is inserted into an operation of moving the node. App., pg. 17. Clearly, removing a node at one location and adding the same node at another location, and vice versa, can be interpreted as moving that node. Figures 5 and 6 provide detailed algorithms for interpreting more complicated sequences of operations as operations of moving a node. App., pg. 17-20. Furthermore, the application clearly details how the operation sequences may be read from memory. "The generated list L of the operation sequences is temporarily stored in memory means such as the main memory 103. Then, the difference computation unit 20 analyzes the list L stored in the main memory 103 to detect MoveNode as shown in Figures 3 to 6." App., pg. 16, ln. 23-27.

For at least these reasons, claims 3 and 14 are believed allowable. The Applicant respectfully request reconsideration and allowance of claims 3 and 14.

Claim 5:

Claim 5 was rejected under 35 USC §112 because of the phrase "...a second step of reading the operation sequences from the memory and changing operations in the operation sequence that are interpreted as a movement of a component into an operation of moving the component." OA, pg. 4.

The Applicants respectfully submit that claim 5 does not contain the language referred to in the Office Action. Thus, the rejection of claim 5 under 35 USC §112 is believed to be traversed.

Claim 10:

Claim 10 recites, in part, "...first processing of, when the structured document having the element addressed by predetermined addressing information..." App., pg. 33, ln. 4-6. The examiner alleges several issues with the cited claim wording.

The Examiner alleges that the cited claim wording "fails to distinctly claim the subject matter..." OA, pg. 4. The Applicants respectfully submit that the Examiner's reasoning is purely conclusory without supporting evidence cited. The Office Action has not explained, and it not evident, why a person of ordinary skill in the art would have found it difficult to distinguish between the uses of the words.

The Examiner further alleges that the cited claim wording "fails to provide one of ordinary skill in the art with a succinct step to perform the claimed invention." OA, pg. 4. The Applicants respectfully submit that the Examiner's reasoning is purely conclusory without supporting evidence. The Office Action has not explained, and it not evident, why a person of ordinary skill in the art would have difficulty in recreating the present invention.

To the contrary, the first processing step of Claim 10 includes specific steps to be performed. The paragraph reads, "...inputting the structured document to analyze the modification and storing an analysis result in a memory..." App., pg. 33, ln. 6-8. As previously discussed in regards to Claim 1, Figure 1 discusses numerous input devices which may facilitate inputting the structured document and a specific method for inputting the structured document. The application also provides detail about the analysis of the modification. The discussion of Figure 2 specifies that "...the document analysis unit 10 analyzes the structured documents and converts them into data in a tree-structure such as a DOM tree..." App., pg. 12, ln. 14-16. The Applicants respectfully submit that the application provides sufficient detail to provide one of ordinary skill in the art with succinct steps to perform the present invention.

Claim 10 further recites, in part, "second processing of reading the analysis result from the memory; and updating the addressing information according to the analyzed modification so that the addressing information addresses at least..." App., pg. 33, ln. 9-12. The Examiner states that the wording is "vexatious." OA, pg. 5.

The Applicants respectfully submit that the Examiner's reasoning is purely conclusory without supporting evidence. The Office Action has not explained, and it not evident, why a person of ordinary skill in the art would have difficulty understanding the cited claim wording. Furthermore, the wording deemed objectionable by the Examiner is substantially similar to wording previously discussed above in regards to Claim 1. Thus, the details provided in the application and discussed above, concerning how to implement the corresponding wording in Claim 1, are equally applicable to the cited wording of Claim 10.

The Examiner further states that the wording "leaves one of ordinary skill in the art with no idea of who or what is reading the information the analysis result and where the result is being read to." OA, pg. 5. The Applicants respectfully submit that no legal requirement exists to specify the actor in a program claim such as Claim 10. Thus, it is irrelevant whether the identity and nature of the actor are specified by the claim.

For at least these reasons, claim 10 is believed allowable. The Applicant respectfully request reconsideration and allowance of claim 10.

Claim 17:

Claim 17 recites, in part, "...difference computation means for computing, when the HTML/XML document for which the annotation data has been made is modified, a difference between an unmodified version and a modified version of the HTML/XML document..." App., pg. 35, ln. 23-27. The examiner alleges several issues with the cited claim wording.

The Examiner alleges that the cited claim wording "includes incomplete thoughts and sentences. " OA, pg. 5. The Applicants respectfully submit that the Examiner's reasoning is purely conclusory without supporting evidence. The Office Action has not explained, and it not evident, why the claim wording includes incomplete thoughts and sentences. To the contrary, removing the parenthetical phrase from the cited wording yields "...difference computation means for computing...a difference between an unmodified version and a modified version of the HTML/XML document..." App., pg. 35, ln. 23-27. This is a complete thought because it clearly indicates the purpose of the difference computation means. The addition of a parenthetical phrase cannot make a thought less complete. Furthermore, the cited wording is a component of an annotation server, as indicated by the preceding wording, "the annotation server comprising:" App., pg. 35, ln. 21-22. Such a component does not need a complete sentence to be fully described; a noun suffices to describe the component.

The Examiner further alleges that the wording is "vexatious in nature because the steps are not complete." OA, pg. 5. The Applicants respectfully submit that the Examiner's reasoning is purely conclusory without supporting evidence. The Office Action has not explained, and it not evident, why the steps are not complete. To the contrary, significant detail is provided concerning how to implement the steps discussed by the cited claim wording. Figure 15 provides numerous details of an implementation of an annotation server. App., pg. 25-26. Figure 15 further specifies that the "annotation server 1500 has functions corresponding to the...the difference computation unit 20, and..." App., pg. 25, ln. 29-30 and pg. 26, ln. 1. This wording suggests that the difference computation unit may be used as the difference computation means of Claim 17. Figures 2 through 6 describe the difference computation unit and the algorithms it applies to determine the difference between structured documents. App., pg. 12-20.

The Examiner further alleges that "[o]ne of ordinary skill in the art would not have a proficient means of understanding or carrying out applicant's invention based upon the presently claimed language." The Applicants respectfully submit that one of ordinary skill in the art could apply the previously cited teachings of Figures 15 and Figures 2 through 6 in order to understand and carry out the present invention.

For at least these reasons, claim 17 is believed allowable. The Applicant respectfully request reconsideration and allowance of claim 17.

CLAIM REJECTIONS UNDER 35 USC §103:

Claims 1-22 were rejected under 35 USC §103 as allegedly obvious over U.S. Patent Application Publication No. US 2002/0054090 ("Silva") in view of U.S. Patent No. 6,785,673 issued to Fernandez et al. ("Fernandez").

A *prima facie* case for obviousness can only be made if the combined reference documents teach or suggest all the claim limitations. MPEP 2143.

Claim 1:

Claim 1 recites, in part, "...updating the addressing information according to the analyzed modification so that the addressing information addresses at least one corresponding element or corresponding elements in the modified structured document." App., pg. 30, ln. 12-16. A notable feature of Claim 1 is that the addressing information itself is updated, as opposed to updating the document using unchanged address information.

"All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

The Applicants respectfully submit that the Examiner has inappropriately disregarded the claim words and simply provides a reference describing addressing information in structured documents. With respect to the teachings of Silva, the patent appears directed toward utilizing address information to extract information from documents for display on thin devices. The Applicants respectfully submit that Silva does not disclose or suggest updating addressing information. The failure to disclose or suggest updating address information in the specific passages cited by the Examiner is substantiated in detail below.

In response to Claim 1, the Office Action alleges that Silva teaches "[a] method of generating robust Xpath [sic] expressions and selecting a node in transcodable markup transcoders". OA, pg. 5. In support of this position, the Office Action cites paragraphs 0048, 0034 and 0059 of Silva. *Id.* The specific citation offered by the Office Action states,

> [0048] Since similar techniques are used to generate a robust XPath expression for both a column-wise and row-wise layout, only XPath generation for the latter will be described. For both, the GUI can generate robust clippings in XPath--**clippings that survive significant changes to the HTML structure** that leave the form of the table of interest form alone. For instance, the table can move from being top-level to being nested several levels deep in another table or vice versa and the XPath expression will continue to work properly. The following form of XPath expression is used for row-wise layout with a user-specified first row:
>
> (//table/tr/td[contains(string(),`<USER-SPECIFIED-LABEL>`) and not(descendant::table)]/parent::tr) [1] (3)
>
> [0034] As can be noted, these expressions can be complicated, and writing them can be an involved task. In addition, there are multiple ways to specify a particular page component, and some may be preferable to others in terms of robustness. Since the Web view system is directed towards the naive user, it is unlikely that he would be able to specify XPath expressions. **Accordingly, a point-and-click graphical user interface (GUI) that lets users select portions of Web pages (as they see them in the Web browser) and automatically generate extraction expressions** is a superior methodology for extracting components from a document. The point-and-click interface provides users with different levels of abstraction corresponding to a breadth-first search in the portion of the document tree that is visible in the browser. For example, if a user is interested in particular cells of a table, he must first select the table and then zoom into the table to select the desired cells. Additional detail on how the GUI produces XPath expressions and other clipping information is provided hereinafter.
>
> [0059] It should be noted that some Web content is available by phone already. Also, some voice browsers and HTML-to-VoiceXML transcoders have been built. However, the effectiveness of such systems is compromised in the presence of improperly structured documents. Furthermore, much of the information on a Web page is not related to

> the primary purpose of the person browsing to the page (e.g., ads, links to other parts of the site). Listening to such pages transcoded into voice is thus usually not a pleasant experience, thereby substantially reducing the utility of such browsers. By simplifying the content retrieval process, and filtering out uninteresting components of Web pages, the Web view architecture of the present invention can render the desired information in voice in a terser, far more user-friendly manner than more general voice browsers can. Silva, para. 0048, 0034, 0059 (emphasis added.)

The Applicants respectfully submit that the cited passage of Silva does not disclose or suggest updating addressing information. To the contrary, Silva discusses methods for generating addressing information which may not require modification even in the event the structure of the document is modified. As the citation offered by the Examiner states, "...the GUI can generate robust clippings in XPath--clippings that survive significant changes to the HTML structure that leave the form of the table of interest form alone." Silva, para. 0048. The citation implies that the addressing information does not require modification because it is versatile enough to refer to the addressed element or elements even in the event of specific, foreseeable changes to the document structure.

The Office Action further alleges that Silva teaches "[s]electing at least two Xpath expression generation methods each method having a configuration for producing Xpath expressions which uniquely identify nodes in transcodable markup and generating a set of Xpath expressions for the selected node using the furthest selected Xpath expression generation methods". OA, pg. 6. In support of this position, the Office Action cites paragraphs 0034, 0047 and 0057 of Silva. *Id.* The specific citation offered by the Office Action states,

> [0034] As can be noted, these expressions can be complicated, and writing them can be an involved task. In addition, there are multiple ways to specify a particular page component, and some may be preferable to others in terms of robustness. Since the Web view system is directed towards the naive user, it is unlikely that he would be able to specify XPath expressions. **Accordingly, a point-and-click graphical user interface (GUI) that lets users select portions of Web pages (as they see them in the Web browser) and automatically generate extraction expressions** is a superior methodology for extracting components from a document. The point-and-click interface provides users with different levels of abstraction corresponding to a breadth-first search in the portion of the document tree that is visible in the browser. For example, if a user is interested in particular cells of a table, he must first select the table and then zoom into the table to select the desired cells. Additional detail on how the GUI produces XPath expressions and other clipping information is provided hereinafter.

[0047] If a user chooses a non-tabular entity (e.g., a paragraph), the GUI asks for predecessor and successor text. The system then determines whether the predecessor (successor) text is within the selected page section or not. Using this information, an XPath expression is automatically generated. If the user chooses a table, the GUI asks him to specify the first row of the table that he is interested in, whether that row contains column labels, and if so, which column labels are of interest. Identifying the first row of interest within the table allows clipping to eliminate elements within the table that are of no interest (e.g., links included in the table simply for layout purposes). FIG. 4 shows a Yahoo! Car page 401 listing used cars in New Jersey. For this exemplary page, the user might specify the row containing DATE as the first row of interest, that that row contains column labels, and that the columns MAKE/MODEL, YEAR, PRICE, and MILES are of interest. The GUI also allows the user to optionally specify a phrase immediately prior to the chosen row (e.g., the word "Showing" in FIG. 4). The user may similarly specify a phrase in the text immediately past the last row he is interested in clipping. If none is specified, the system assumes that all rows are of interest. The GUI also asks him to identify the row labels that are of interest (if any) and whether the table is laid out row-wise (e.g., as in Yahoo! stock quotes) or column-wise (e.g., as in Quicken stock quotes). This information is very valuable for transcoding data for output to a small screen device or a telephone.

[0057] FIG. 6 is a flowchart that summarizes the steps for accessing and replaying a Web view from a specified type of device, in accordance with the present invention. At step 601, the user, from his Web client, sends a request to the Web view server for a particular Web view. Included within the request is what type of device the Web client is. Also, if the requested Web view is parameterized, the parameters are supplied as part of the request. The type of device information can be encoded in the URL or in a GET or POST statement as a parameter. The parameterized parameters could also be provided to the Web view server through an encoded URL or in a form interface that is provided to back to the user's device when the Web view URL is entered. The user would then fill in each of these parameters in the form and forward them to the server. The request may go through a transcoding proxy. It should be noted that the user who accesses a personalized Web view from the Web view server does not necessarily have to be the same person who created the Web view. At step 602, the Web view server retrieves the requested Web view from its Web view database, filling in any parameters supplied by the user as it replays the Web view through the recorded series of navigation steps. At step 603, once the final page is reached, the Web view server applies the recorded XPath expressions in the Web view specification that are applicable to the device specified in the request. The resulting content is then further processed to remove uninteresting content (if any) by traversing the parsed document representation and including only the interesting parts of the tree to the document being generated. At step 604, the resulting generated document is returned to the Web client. When it is returned to the client, the Web view may go through a proxy, which transcodes the clipped content into a format acceptable to the requesting client. Thus, the Web view returned from the Web view server is transcoded into whatever language that is supported by the Web client in the device. Silva, para. 0034, 0047, 0057 (emphasis added.)

The Applicants respectfully submit that the cited passage of Silva likewise fails to disclose or suggest updating addressing information. While the citation discusses utilizing addressing information such as XPath to transcode an original document, it does not discuss updating addressing information in response to modifications to the original document. Indeed, Silva provides no teaching to update addressing information in response to changes in the original document because transcoding a document generally does not modify the original document.

The Office Action further alleges that Fernandez teaches "the use of redundant set of Xpath [sic] expressions and an intermediate representation of XML view queries called a view tree has been created that is general enough to express the XML mappings in any of these system [sic]." OA, pg. 6. In support of this position, the Office Action cites col. 3, ln. 38-54 of Fernandez. *Id.* The specific citation offered by the Office Action states,

> A query language according to another aspect of the present invention, can be adapted for operation with a variety of systems. For example, the query language can express the transformations expressible in existing XML publishing tools, such as those provided by relational database systems. For example, the IBM DB2 XML Extender provides a Data Access Definition (DAD) language, Microsoft SQL Server has an XML view-definition module, and the Oracle XML SQL Utility exports relational data in a fixed, canonical XML view. In another aspect of the present invention, an intermediate representation of XML view queries called a view tree has been created **that is general enough to express the XML mappings in any of these systems.** An illustrative algorithm of the invention takes a view tree as input, and therefore could be directly applied to the XML view definitions expressed by these tools. Fernandez, col. 3, ln. 38-54 (emphasis added.)

This citation also fails to disclose or suggest updating addressing information. The view tree mechanism taught by Fernandez is not equivalent to, nor does it suggest, changing address information in accordance with modifications to a structured document.

For at least these reasons, claim 1 is believed allowable over the cited art. The Applicant respectfully request reconsideration and allowance of claim 1.

<u>Claim 2:</u>

Claim 2 is dependent on Claim 1 and recites, "A data processing method according to Claim 1, wherein the step of updating the addressing information comprises updating the addressing information written in XPath." App., pg. 30, ln. 17-20. It is noted that Claim 2 specifically comprises updating addressing information. It includes the words "updating the addressing information". *Id.* Furthermore, Claim 2 is dependent on Claim 1, which as previously discussed comprises updating addressing information.

In response to Claim 2, the Office Action alleges that Silva teaches "[r]esolving each Xpath expression in the set, the resolutions producing a set of candidate nodes; and identifying the selected node from among the set of candidate nodes based upon the set of Xpath expressions in the set resolving to the selected node more than any other node in the set of candidate nodes." OA, pg. 6. In support of this position, the Office Action cites paragraphs 0042 and 0048 of Silva. *Id.* The specific citation offered by the Office Action states,

> [0042] Extraction expressions also need to be made robust to changes to Web pages. For example, in the XPath expression (1) above, if the position of the center tag containing the desired tables changes (e.g., a new preceding sibling center tag appears in the document), the expression will no longer retrieve the correct tables. Instead of absolute positions of nodes, **the specification needs to include other information that helps the system uniquely identify components to be extracted, even if the node positions happen to change.** For instance, the XPath expression (2) specifies tables that contain the "Price" or "Option" string--this expression would still retrieve the correct itineraries even if new center tags are added. How these expressions can be automatically generated will be discussed hereinafter.
>
> [0048] Since similar techniques are used to generate a robust XPath expression for both a column-wise and row-wise layout, only XPath generation for the latter will be described. For both, the GUI can generate robust clippings in XPath--**clippings that survive significant changes to the HTML structure** that leave the form of the table of interest form alone. For instance, the table can move from being top-level to being nested several levels deep in another table or vice versa and the XPath expression will continue to work properly. The following form of XPath expression is used for row-wise layout with a user-specified first row:
>
> (//table/tr/td[contains(string(),`<USER-SPECIFIED-LABEL>`) and not(descendant::table)]/parent::tr) [1] (3) Silva, para. 0042 and 0048 (emphasis added.)

The Applicants respectfully submit that the cited passage of Silva does not disclose or suggest updating addressing information. To the contrary, Silva discusses methods for generating addressing information which may not require modification even in the event the structure of the document is modified.

As the citation offered by the Examiner states, "the specification needs to include other information that helps the system uniquely identify components to be extracted, even if the node positions happen to change." Silva, para. 0042. It is noted that the context of the quote clarifies that a "specification" refers to addressing information. The citation implies that the addressing information does not require modification because it is versatile enough to refer to the nodes even in the event of specific, foreseeable changes to the document structure. Furthermore, the reasons why Paragraph 0048 of Silva does not teach updating addressing information discussed above under Claim 1 apply equally to Claim 2.

For at least these reasons, claim 2 is believed allowable over the cited art. The Applicant respectfully request reconsideration and allowance of claim 2.

Claim 3:

Claim 3 recites, in part, "...in which each operation for transforming one of the tree-structured data items into the other tree-structured data item is expressed as a combination of predetermined operations on a component of a tree-structure..." App., pg. 31, ln. 2-6. Claim 3 further recites, in part, "...changing operations in the operation sequence that are interpreted as a movement of a component into an operation of moving the component." App, pg. 31, ln. 9-11.

In response to Claim 3, the Office Action alleges that Silva teaches "[s]electing at least two Xpath expressions generation methods from the group consisting of a tag-specific methodology, an anchoring methodology and a cascading filtering methodology." OA, pg. 6-7. In support of this position, the Office Action cites paragraphs 0045, 0035 and 0065 of Silva. *Id.* The specific citation offered by the Office Action states,

> [0045] To support reliable parameterization of Web views, two issues should preferably be resolved: internal attribute names that are un-descriptive and invalid selections. Consider for example the page at the Travelocity Web site where a user specifies his itinerary details. The internal name for the departure month attribute is depdtmnl, which can be hard to identify. Also month values must have a specific format, e.g., April is represented by "Apr", and if the user inputs "April", the submission will fail. The WebVCR component of the Web view system can be extended to allow users to edit input attributes directly in the Web view. At recording time, users can specify mappings from internal names to more descriptive tags of their choice. In addition, extra information is saved for elements (e.g., all values in a selection list are saved) so that inputs can be checked for validity. This additional

information can be useful for transcoding Web views, to be discussed hereinafter. Checking for validity of value, however, is only possible for elements such as selection lists and radio buttons, where a domain is well defined, and is not possible for text fields. Thus, even though the likelihood of failures can be reduced, they cannot be entirely avoided. It should be noted that parameterization only works reliably for deterministic sites. If there are different navigation paths for different values (or combination of values) of parameters, it will invariably fail when an alternate path is taken.

[0035] With reference again to FIG. 2, a Web view specification for the Travelocity scenario example is shown. As earlier noted, the specification includes a smart bookmark 201 identified by id="juliana_travel", for the "9 Best Itineraries link"at the Travelocity Web cite. The Web view specification further includes an extract data specification 202, which defines what information is to be extracted from the source bookmarked page as a function of the type of device on which the extracted information is to be displayed. The first part of the extraction specification 202, on line 203, points to the smart bookmark 201 that retrieves the desired page. The <EXTRACT> expressions 204 and 205 contain different extraction specifications that may be applied for displaying the extracted information on a PDA device and a WAP telephone, respectively. For example, as shown, if the Web view is to be displayed on a PDA, the first 3 itineraries (the extraction tag in 204 with fragment_name="first.sub.--3_itineraries") are chosen to be displayed. If the Web view is to be displayed in a Web-enabled cellular phone with a 3-line display, only a single itinerary is chosen to be displayed (e.g., the extraction tag in 205 with fragment.sub.13name="firs- t_itinerary"). The <DEVICE> expressions 206 and 207 link the device on which the extracted information is to be displayed with the particular extraction expressions 204 and 205. Thus, as noted in FIG. 2, if the device is, for example, a Nokia model 9000 Web-enabled cellular phone, the extraction expression it is linked with is 205 via the <DISPLAY fragment="first itinerary"> statement. Similarly, if the PDA device is a Palm Pilot, the extraction expression it its linked with is 204 via the <DISPLAY fragment="first.sub.--3_-itineraries"> statement.

[0065] In order to intelligently transcode a table, Web view annotations are helpful. A table may be organized row-wise, column-wise, or neither (e.g., being used simply for layout)--and each requires substantially different transcoding. For instance, if the table in FIG. 4 were treated as being there simply for layout, the transcoded voice would be partially incomprehensible. For example, it would say: "Showing 1-15 of 34 listings Previous Ads Next Ads DATE MAKE MODEL YEAR PRICE FULL LISTING Oct. 26, 2000 Acura Integra 1995 11000," etc. The knowledge that MAKE/MODEL, YEAR, PRICE, and MILES are the columns to transcode and that the table is laid out row-wise allows the transcoder to pair headers and values together (as defined in FIG. 8), and to eliminate uninteresting data (e.g., DATE, FULL LISTING columns). Similar techniques are useful for small screens (e.g., on WAP phones) even where tables are supported, because they produce something much more readable than tables with rows that wrap lines. Silva, para. 0045, 0035, 0065.

The Applicants respectfully submit that the cited passage of Silva does not disclose or suggest expressing a set of operations for transforming a first tree-structured data item into a second tree-structured data item as a combination of predetermined operations on a component of a tree-structure. Furthermore, the cited passage of Silva does not disclose or suggest changing operations in the operation sequence that are interpreted as a movement of a component into an operation of moving the component.

[redacted]

Best Available Copy

The Applicants respectfully submit that the cited passage of Silva does not disclose or suggest expressing a set of operations for transforming a first tree-structured data item into a second tree-structured data item as a combination of predetermined operations on a component of a tree-structure. Furthermore, the cited passage of Silva does not disclose or suggest changing operations in the operation sequence that are interpreted as a movement of a component into an operation of moving the component.

For at least these reasons, claim 3 is believed allowable over the cited art. The Applicant respectfully request reconsideration and allowance of claim 3.

Claim 4:

Claim 4 is dependent on Claim 3 and recites, "A difference computation method according to Claim 3, wherein the first step comprises creating an operation sequence in which each operation for transforming the tree-structured data is expressed as a combination of operations of inserting, removing, or modifying a node or a subtree of a tree structure." App., pg. 31, ln. 12-17.

In rejecting Claim 4, the Office Action alleges that Silva teaches "[s]electing at least two Xpath expressions generation methods each method having a configuration for producing Xpath expressions which uniquely identify nodes in transcodable markup and generating a set of Xpath expressions for the selected node using the furthest selected Xpath expression generation methods". OA, pg. 7. In support of this position, the Office Action cites paragraphs 0034, 0047 and 0057 of Silva. *Id.* The specific citation offered by the Office Action was already reproduced in the discussion of Claim 1.

The Applicants respectfully submit that the cited passage of Silva does not disclose or suggest expressing an operation sequence as a combination of operations of inserting, removing, or modifying a node or a subtree of a tree structure.

The Office Action further alleges that Fernandez teaches "the use of redundant set of Xpath expressions and an intermediate representation of XML view queries called a view tree has been created that is general enough to express the XML mappings in any of these system. [sic]" OA, pg. 7. In support of this position, the Office Action cites col. 3, ln. 38-54 of Fernandez. *Id.* The specific citation offered by the Office Action states,

> A query language according to another aspect of the present invention, can be adapted for operation with a variety of systems. For example, the query language can express the transformations expressible in existing XML publishing tools, such as those provided by relational database systems. For example, the IBM DB2 XML Extender provides a Data Access Definition (DAD) language, Microsoft SQL Server has an XML view-definition module, and the Oracle XML SQL Utility exports relational data in a fixed, canonical XML view. In another aspect of the present invention, an intermediate representation of XML view queries called a view tree has been created **that is general enough to express the XML mappings in any of these systems.** An illustrative algorithm of the invention takes a view tree as input, and therefore could be directly applied to the XML view definitions expressed by these tools. Fernandez, col. 3, ln. 38-54 (emphasis added.)

This citation also fails to disclose or suggest expressing an operation sequence as a combination of operations of inserting, removing, or modifying a node or a subtree of a tree structure.

For at least these reasons, claim 4 is believed allowable over the cited art. The Applicant respectfully request reconsideration and allowance of claim 4.

Claim 6:

Claim 6 is dependent on Claim 5 and recites, "An addressing information generation system according to Claim 5, further comprising a document analysis unit for analyzing structures of the structured documents and converting the structures into tree-structured data items, wherein the difference computation unit computes the difference by comparing the tree-structured data items corresponding to the structured documents converted by the document analysis unit." App., pg. 32, ln. 1-9. It is noted that Claim 6 specifically comprises computing a difference between two separate data items. It includes the words "computes the difference" and "comparing the tree-structured data items". *Id.* Furthermore, Claim 6 is dependent on Claim 5, which comprises computing a difference between structured documents.

In response to Claim 6, the Office Action alleges that Silva teaches "[a] set of differing Xpath expression generators Xpath expression; an interface, and an Xpath generator processor." OA, pg. 7-8. In support of this position, the Office Action cites paragraphs 0047, 0048 and 0057 of Silva. *Id.* The specific citation offered by the Office Action states,

> [0034] As can be noted, these expressions can be complicated, and writing them can be an involved task. In addition, there are multiple ways to specify a particular page component, and some may be preferable

to others in terms of robustness. Since the Web view system is directed towards the naive user, it is unlikely that he would be able to specify XPath expressions. Accordingly, a point-and-click graphical user interface (GUI) that lets users select portions of Web pages (as they see them in the Web browser) and automatically generate extraction expressions is a superior methodology for extracting components from a document. The point-and-click interface provides users with different levels of abstraction corresponding to a breadth-first search in the portion of the document tree that is visible in the browser. For example, if a user is interested in particular cells of a table, he must first select the table and then zoom into the table to select the desired cells. Additional detail on how the GUI produces XPath expressions and other clipping information is provided hereinafter.

[0048] Since similar techniques are used to generate a robust XPath expression for both a column-wise and row-wise layout, only XPath generation for the latter will be described. For both, the GUI can generate robust clippings in XPath--clippings that survive significant changes to the HTML structure that leave the form of the table of interest form alone. For instance, the table can move from being top-level to being nested several levels deep in another table or vice versa and the XPath expression will continue to work properly. The following form of XPath expression is used for row-wise layout with a user-specified first row:

```
(//table/tr/td[contains(string(),`<USER-SPECIFIED-LABEL>`) and
not(descendant::table)]/parent::tr) [1] (3)
```

[0067] FIG. 9 is a flowchart that summarizes the steps of creating a Web view for VoiceXML applications in accordance with the present invention. At step 901, the user initiates recording of the Web view using a smart bookmark recording applet that is stored in his own desktop machine or that is downloaded from a remote site. At step 902, the starting page of the Web view being created is specified if the final page from which information is to be extracted to form the personalized Web view cannot be reached directly. At step 903, the recorder applet records each of the user's navigation actions as he browses to the final page containing the information to be clipped for the personalized Web view. These navigation actions include, for example, links taken, forms filled out, button clicks, and selections from pull-down menus. At step 904, the user selects the components of interest to be extracted. At step 905, selected components are extracted through a specified XPath expression or through a GUI that generates the expression. Also, depending on the type of the component, the user can insert annotations that will guide the transcoder to generate the appropriate VoiceXML for the component when the Web view is replayed through VoiceXML. Example of these annotations include whether a table should be read row-wise or column-wise, and what labels that are associated with a table are of interest. These annotations are also useful to improve the robustness of the voice view. At step 906, the Web view specification and annotations are saved. At step 907, the Web view specification is uploaded to the Web view server for later replay through the VoiceXML transcoder for output to the user's audio terminal, such as a telephone.

[0032] XPath (see, e.g, http://www.w3.org/TR/xpath for a description of the XPath language) is an example of a mechanism for specifying

> extraction expressions. XPath views an XML document as a tree and provides a flexible mechanism for addressing any node in this tree. One drawback of using XPath, however, is its requirement that pages be well formed. Since browsers are very forgiving in this respect, many Web sites generate pages that are ill formed (e.g., have overlapping tags, missing end tags, etc.). Consequently, the Web view system must first clean up HTML pages (e.g., using tools such as HTML Tidy [see, e.g., http://www.w3.org/People/Raggett/tidy]) before XPath can be applied. Another alternative for specifying extraction expressions is the XML DOM API. However, XPath allows a more flexible and easier way to create robust clipping expressions that are immune to minor changes in page structure. DOM addresses (without storing extra information, or using other heuristics to compensate for page changes) can be very brittle even to minor layout changes. Silva, para. 0034, 0048, 0067 and 0032.

The Applicants respectfully submit that the cited passage of Silva does not disclose or suggest computing a difference between two separate data items.

For at least this reason, claim 6 is believed allowable over the cited art. The Applicant respectfully request reconsideration and allowance of claim 6.

Claims 7 and 8:

In response to Claims 7 and 8, the Office Action alleges that Silva teaches "[g]enerating a set of Xpath expressions for the selected node using the further selected Xpath expression generation methods." OA, pg. 8. In support of this position, the Office Action cites paragraphs 0034, 0048, 0067 and 0032 of Silva. *Id.* The specific citation offered by the Office Action was already reproduced in the discussion of Claim 1.

The Applicants respectfully submit that the cited passage of Silva does not disclose or suggest computing a difference between two separate data items.

In response to Claims 7 and 8, the Office Action further alleges that Silva teaches "[r]esolving each Xpath expression in the set, the resolutions producing a set of candidate nodes; and identifying the selected node from among the set of candidate nodes based upon the set of Xpath expressions in the set resolving to the selected node more than any other node in the set of candidate nodes." OA, pg. 8. In support of this position, the Office Action cites paragraphs 0042 and 0048 of Silva. *Id.* The specific citation offered by the Office Action was already reproduced in the discussion of Claim 1.

The Applicants respectfully submit that this second cited passage of Silva does not disclose or suggest computing a difference between two separate data items.

For at least these reasons, claims 7 and 8 are believed allowable over the cited art. The Applicant respectfully request reconsideration and allowance of claims 7 and 8.

Claim 9:

Claim 9 is dependent on Claim 8 and recites, "An addressing information generation system according to Claim 8, wherein the addressing information generation unit generates an XPath for the other structured document by regenerating LocationSteps forming an XPath for the particular structured document based on the difference between the structured documents and on the XPath for the particular structured document." App., pg. 32, ln. 20-27. Because Claim 8 is dependent on Claim 5, Claim 9 is indirectly dependent on Claim 5. It is noted that Claim 9 specifically comprises regenerating LocationSteps forming an XPath.

In response to Claim 9, the Office Action alleges that Silva teaches "[r]esolving each Xpath expression in the set, the resolutions producing a set of candidate nodes; and identifying the selected node from among the set of candidate nodes based upon the set of Xpath expressions in the set resolving to the selected node more than any other node in the set of candidate nodes." OA, pg. 8. In support of this position, the Office Action cites paragraphs 0042 and 0048 of Silva. *Id.* The specific citation offered by the Office Action was already reproduced in the discussion of Claim 1.

The Applicants respectfully submit that this cited passage of Silva does not disclose or suggest regenerating LocationSteps forming an XPath. Indeed, LocationSteps are not specifically mentioned in the cited passage.

For at least this reason, claim 9 is believed allowable over the cited art. The Applicant respectfully request reconsideration and allowance of claim 9.

Claims 10-23:

Claims 10-23 were rejected for the same reasons as claims 1-9. OA, pg. 8. Since claims 1-9 are believed allowable, claims 10-22 are also believed allowable for at least the same reasons as claims 1-9.

RECEIVED
CENTRAL FAX CENTER
JUL 06 2006

If the rejections of claims 10-23 are maintained in subsequent office actions, the Applicants kindly request a more detailed explanation of why the elements and limitations of claims 10-23 are not believed patentable over the cited references.

Claim 23:

The original claim listing contained two claims numbered Claim 22. By this amendment, the second occurrence of Claim 22 is renumbered to Claim 23.

CONCLUSION

In view of the forgoing remarks, it is respectfully submitted that this case is now in condition for allowance and such action is respectfully requested. If any points remain at issue that the Examiner feels could best be resolved by a telephone interview, the Examiner is urged to contact the attorney below.

Please charge Deposit Account 50-0510 the amount of $50 for one additional dependent claim, and $120 for a one-month extension for the period for response to the Office Action. No other fee is believed due with this Amendment, however, should a fee be required please charge Deposit Account 50-0510. Should any extensions of time be required, please consider this a petition thereof and charge Deposit Account 50-0510 the required fee.

Respectfully submitted,

Ido Tuch

Dated: July 6, 2006

Ido Tuchman, Reg. No. 45,924
Law Office of Ido Tuchman
82-70 Beverly Road
Kew Gardens, NY 11415
Telephone (718) 544-1110
Facsimile (718) 544-8588